



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

کارشناسی ارشد علوم کامپیوتر گرایش داده کاوی

پروژه شماره یک درس داده کاوی

نگارش

حدیث حق شناس جزی

استاد راهنما

مهدی قطعی

استاد مشاور

بهنام یوسفی مهر

مهر ۱۴۰۱

## چکیده

در این گزارش به معرفی پایتون و چند زبان برنامه نویسی دیگر و علت برتری پایتون اشاره شده است. تمرکز این گزارش بر معرفی این نرم افزار و سپس معرفی کتابخانه های پایتون و استفاده آن ها در حل مسائل مرتبط با کتاب درس داده کاوی (Data mining concepts and techniques) می باشد. در ادامه و به عنوان تمرین به حل یک مسئله پیشبینی انرژی به روش رگرسیون با یکی از مهمترین کتابخانه های پایتون خواهیم پرداخت.

چکیده..... ۲

فصل اول معرفی پایتون..... ۴

۱-۱ پایتون چیست؟..... ۴

۱-۲ چرا پایتون؟..... ۵

۱-۳ کتابخانه های پایتون..... ۶

۱-۴ معرفی چند کتابخانه کاربردی..... ۷

۱-۵ حل مسائل کتاب درسی با پایتون..... ۹

فصل دوم پایتون در مقایسه با دیگر نرم افزار ها..... ۱۰

۲-۱ مقایسه کاربرد پایتون و دیگر نرم افزارها در میان تحلیلگران داده..... ۱۰

۲-۲ معرفی نرم افزار R..... ۱۱

۲-۳ معرفی نرم افزار Rapidminer..... ۱۲

فصل سوم مسئله پیشبینی انرژی..... ۱۳

۳-۱ حل مسئله پیش بینی انرژی با روش رگرسیون خطی..... ۱۳

جمع بندی و نتیجه گیری..... ۱۶

منابع و مراجع..... ۱۷

## ۱ فصل اول

### ۱-۱ پایتون چیست؟

پایتون یک زبان برنامه نویسی شی گرا، تفسیری و سطح بالا است که توسط Guido van Rossum طراحی و در سال ۱۹۹۱ منتشر شد. فلسفه اصلی طراحی پایتون "خوانایی بالای کد" است. ساختار زبانی و دیدگاه شیء گرا در پایتون به گونه ای طراحی شده است که به برنامه نویس امکان نوشتن کد منطقی و واضح (بدون ابهام) را برای پروژه های کوچک و بزرگ میدهد. برخلاف برخی زبانهای برنامه نویسی رایج دیگر که بلاکهای کد در آکولاد تعریف میشوند (به ویژه زبانهایی که از نحو زبان سی پیروی میکنند) در زبان پایتون از نویسه فاصله و جلو بردن متن برنامه برای مشخص کردن بلاکهای کد استفاده میشود، بدین ترتیب بلاکهای کد به صورت خودکار ظاهری مرتب دارند. کلمات کلیدی و اصلی این زبان به صورت حداقلی تهیه شده اند و در مقابل کتابخانه هایی که در اختیار کاربر است بسیار وسیع هستند. این برنامه در زمینه های مختلف زیادی مانند توسعه وب، توسعه بازی، توسعه نرم افزار، یادگیری ماشین و هوش مصنوعی، علم داده و تجسم داده ها، انجام ریاضیات پیچیده، رابط کاربری گرافیکی دسکتاپ، برنامه های اسکراپینگ وب و برنامه های صوتی و تصویری کاربرد دارد.

## ۲-۱ چرا پایتون؟

- پایتون روی پلتفرم‌های مختلف (ویندوز، مک، لینوکس و غیره) کار می‌کند.
- پایتون یک نحو (syntax) ساده شبیه به زبان انگلیسی و محاوره دارد و خوانا است. یکی از دلایل محبوبیت پایتون همین موضوع می‌باشد زیرا افراد مبتدی ارتباط بهتری با پایتون نسبت به سایر زبان‌های برنامه‌نویسی برقرار میکنند و یادگیری آن ساده‌تر است.
- پایتون به گونه‌ای طراحی شده است که به برنامه‌نویس‌ها اجازه می‌دهد تا برنامه‌هایی با خطوط کمتر نسبت به برخی از زبان‌های برنامه‌نویسی دیگر بنویسند.
- پایتون بر روی یک سیستم‌مفسر اجرا می‌شود، به این معنی که کد را می‌توان به محض نوشتن اجرا کرد و این ویژگی اشکال زدایی را آسان‌تر و مدیریت حافظه را کارآمدتر میکند. (پایتون از مدیریت خودکار حافظه استفاده میکند)
- پایتون از خطوط جدید برای تکمیل یک دستور استفاده می‌کند، برخلاف سایر زبان‌های برنامه‌نویسی که اغلب از نقطه ویرگول یا پرانتز استفاده می‌کنند.
- پایتون برای تعریف محدوده به تورفتگی، متکی است. مانند محدوده حلقه‌ها، توابع و کلاس‌ها. سایر زبان‌های برنامه‌نویسی اغلب از آکولادها برای این منظور استفاده می‌کنند.
- پایتون متن‌باز است. یعنی تحت مجوز منبع باز OSI توسعه یافته است و آن را قابل استفاده و اصلاح و بهبود میکند یعنی شما این آزادی را دارید که کد پایتون را برای بهبود نیازهای خود بدون مواجهه با هیچ عواقبی تغییر دهید.
- یکی دیگر از ویژگی‌های مهم پایتون کتابخانه‌های زیاد آن است که برنامه‌نویسی را بسیار راحت کرده و شما می‌توانید از کدهای افرادی دیگر استفاده کنید.

### ۳-۱ کتابخانه های پایتون :

در پایتون برای راحتی کار برنامه نویسان ساختارهایی به نام کتابخانه ایجاد شده است . کتابخانه‌های پایتون کدها و توابع مفیدی هستند که توسط سایر برنامه نویسان ایجاد شده و به صورت پکیج یا کتابخانه درآمده تا سایرین از این کدها در برنامه خود استفاده کنند و چرخ را از اول اختراع نکنند. همین کتابخانه های گسترده پایتون است که کار برنامه نویسی را بسیار راحت و سریع می‌کند.

در پایتون دو نوع کتابخانه وجود دارد. نوع اول کتابخانه‌هایی است که به همراه خود پایتون روی سیستم نصب می‌شود که بخشی از این کتابخانه‌ها مربوط به هسته زبان پایتون است و بخش دیگر کتابخانه‌هایی هستند که جزء هسته آن نیستند و باید به کمک عبارت `import` این کتابخانه‌ها را به برنامه اضافه کرد و از توابع آن استفاده کرد.

همچنین در پایتون توابع از پیش تعریف شده ای هستند که به دلیل استفاده و کاربرد زیاد به پایتون اضافه شده اند که به آنها `built-in function` گفته می‌شود.

## ۴-۱ معرفی چند کتابخانه کاربردی :

### کتابخانه Sickit-learn :

اسکیت لرن (Scikit-learn) یکی از کتابخانه های محبوب حوزه ماشین لرنینگ در پایتون است. این کتابخانه شامل ابزارهای ساده و کارآمد برای داده کاوی و تحلیل داده است از کاربردهای اسکیت لرن می توان به موارد زیر اشاره کنیم:

- طبقه بندی: مشخص کردن اینکه یک شی متعلق به کدام دسته است
- رگرسیون: پیش بینی یک ویژگی با ارزش مداوم در ارتباط با یک شی
- خوشه بندی: گروه بندی اتوماتیک شی ها شبیه به هم درون یک مجموعه
- کاهش ابعاد: کاهش تعداد متغیرهای تصادفی برای رسیدگی کردن
- انتخاب مدل: مقایسه، اعتبار سنجی و انتخاب پارامترها و مدل ها
- پیش پردازش: استخراج ویژگی ها و نرمال سازی

### کتابخانه Numpy :

نامپای یا numpy یک کتابخانه پایتون است که برای کار با آرایه ها به وجود آمده است. کتابخانه numpy توابعی برای انجام عملیات های گوناگون در جبر خطی، تبدیل فوریه و ماتریس ها دارد. نامپای در سال ۲۰۰۵ و به صورت یک پروژه متن باز (open source) ایجاد شد Numpy. سرواژه ی عبارت Numerical Python به معنای پایتون عددی یا پایتون محاسباتی است.

علت به وجود آمدن این کتابخانه این بود که در پایتون چیزی به عنوان آرایه وجود ندارد؛ با این حال می توان از لیست (list) به عنوان آرایه استفاده کرد. مشکل لیست آن است که سرعت پردازش داده ها در آن بسیار پایین است Numpy. تلاش دارد شیئی را به عنوان آرایه ارائه دهد که ۵۰ برابر از لیست سریع تر است. شیئی که به عنوان آرایه در numpy موجود است، ndarray نام دارد. نامپای توابع زیادی دارد که کار با ndarray را بسیار راحت کرده اند. توجه داشته باشید که با توجه به استفاده ی گسترده از آرایه ها در علوم داده و با توجه به حجم بودن داده ها، سرعت مقوله ی بسیار مهمی میباشد.

## کتابخانه Pandas :

پانداس یک کتابخانه متن باز است که کارایی بالا، ساختاری با قابلیت استفاده آسان و ابزارهای تحلیل داده برای برنامه نویسی پایتون را فراهم می کند. پانداس یک کتابخانه قدرتمند برای تحلیل و پردازش داده ها است. امروزه از pandas به طور گسترده در مباحث علم داده، یادگیری ماشین و یادگیری عمیق استفاده می شود. عمده کاربران pandas، دانشمندان علم داده و تحلیل گران اقتصادی مانند تحلیل گران بورس هستند.

از ویژگی های این کتابخانه در حوزه علوم داده میتوان به موارد زیر اشاره کرد:

- انجام محاسبات آماری روی داده ها مانند محاسبه بزرگترین و کوچکترین مقدار، محاسبه میانگین داده ها، محاسبه صدک ها و الی آخر.
- نگاهی به نحوه توزیع داده ها در یک ستون
- بررسی احتمال وابسته بودن ستون ها به یکدیگر
- پاک سازی داده ها: مثلاً حذف کردن ردیف هایی که مقادیر ناقص دارند و یا حذف کامل بخش هایی که مقداری ندارند و خالی هستند، یا مرتب سازی یا فیلتر کردن ستون های خاص بر اساس شرط هایی خاص و الی آخر.
- همکاری با پکیج های بزرگ دیگر مانند Matplotlib برای بصری سازی داده ها: تولید نمودار های مختلف، هیستوگرام ها و الی آخر.
- ذخیره سازی داده های پاک سازی شده در یک فایل اکسل و ...



۵-۱ حل مسائل کتاب درسی ( غیر از فصل های حذفی ۴ و ۵ ) :

فصل ۱ و ۲ : توضیحات کلی است و نیازی به کتابخانه ها ندارد

فصل ۳ : پیش پردازش یا preprocessing به کمک کتابخانه های numpy و pandas انجام میپذیرد.

فصل ۶ و ۷ : شناسایی الگو های پرتکرار یا identifying frequent patterns به کمک کتابخانه PAMI انجام میپذیرد.

فصل ۸ و ۹ : طبقه بندی یا classification به کمک کتابخانه scikit-learn انجام میپذیرد.

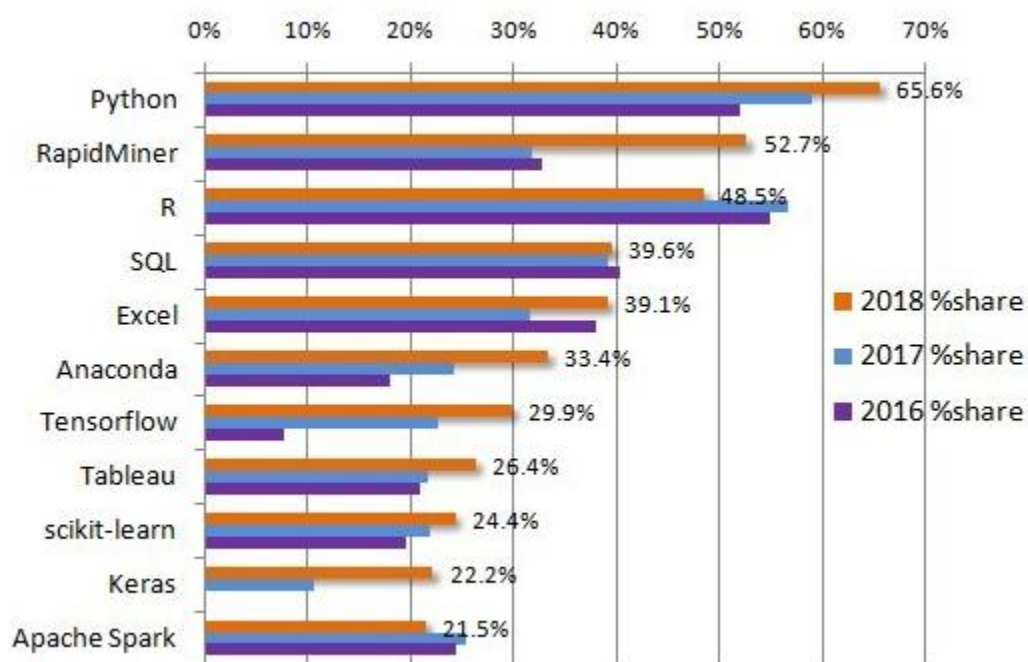
فصل ۱۰ و ۱۱ : خوشه بندی یا clustering به کمک کتابخانه های PyCaret و scikit-learn انجام میپذیرد.

فصل ۱۲ : داده های پرت یا Outlier data به کمک کتابخانه های PyOD و alibi-detect و PyNomaly انجام میپذیرد.

## ۲ فصل دوم

### ۲-۱ مقایسه پایتون با زبان های برنامه نویسی دیگر:

در زیر با استناد به نمودار، به میزان محبوبیت و میزان رشد محبوبیت پایتون در میان تحلیلگران داده نگاهی می اندازیم و با معرفی و مقایسه کوتاهی از دیگر برنامه های محبوب میان تحلیل گران داده خواهیم داشت.



با توجه به نمودار بالا محبوبیت پایتون در میان برنامه نویسان داده افزایش چشم گیری داشته است. اما در جایگاه های بعدی برنامه های R و RAPIDMINER و ... حضور دارند که در زیر به توضیح مختصری از آنها میپردازیم. بدیهی است که در پروژه های تحلیل داده گاها چند نرم افزار در کنار یکدیگر مورد استفاده قرار گرفته و نرم افزار هایی که در زیر به آنها اشاره میکنیم نیز در موارد مشابه کاربردهای فراوان دارند.

## ۲-۲ معرفی نرم افزار R

R یک زبان برنامه‌نویسی منبع باز می‌باشد که برای تجزیه و تحلیل آماری و تجسم داده بهینه شده است. R که در سال ۱۹۹۲ توسعه یافت، دارای یک اکوسیستم غنی با مدل‌های داده پیچیده و ابزارهای ظریف برای گزارش داده است. در آخرین شمارش، بیش از ۱۳ هزار بسته R از طریق شبکه آرشیو جامع R برای تجزیه و تحلیل عمیق در دسترس بود.

R در میان دانشمندان و محققان علوم داده محبوب است و طیف گسترده‌ای از کتابخانه‌ها و ابزارها را برای موارد زیر فراهم می‌کند: پاک کردن و آماده سازی داده، ایجاد تجسم داده، آموزش و ارزیابی الگوریتم‌های یادگیری ماشین و یادگیری عمیق.

R معمولاً در RStudio، یک محیط توسعه یکپارچه (IDE) برای تجزیه و تحلیل آماری ساده، تجسم و گزارش استفاده می‌شود. از برنامه‌های R می‌توان به طور مستقیم و تعاملی در وب استفاده کرد.

### تفاوت اصلی بین R و پایتون

- R به طور عمده برای تجزیه و تحلیل آماری استفاده می‌شود در حالی که پایتون رویکرد کلی‌تری را به علم داده ارائه می‌دهد.
- هدف اصلی R تجزیه و تحلیل داده و آمار است در حالی که هدف اصلی پایتون استقرار و تولید می‌باشد.
- کاربران R عمدتاً متشکل از محققان و متخصصان تحقیق و توسعه هستند در حالی که کاربران پایتون را بیشتر برنامه‌نویسان و توسعه‌دهندگان تشکیل می‌دهند.
- R انعطاف‌پذیری را برای استفاده از کتابخانه‌های موجود و پایتون انعطاف‌پذیری را برای ساخت مدل‌های جدید از ابتدا فراهم می‌کند.
- یادگیری R در آغاز دشوار است در حالی که پایتون خطی بوده و یادگیری آسانی دارد.
- R برای اجرای محلی یکپارچه است در حالی که پایتون به خوبی با برنامه‌ها ادغام شده است.
- R و Python می‌توانند حجم عظیمی از پایگاه داده را مدیریت کنند.

- R را می‌توان در R Studio IDE و Python را در Spyder و Ipython Notebook IDE استفاده کرد.

### ۲-۳ معرفی نرم افزار RapidMiner

RapidMiner یک نرم افزار داده کاوی است که توسط کمپانی به همین اسم تولید شده است. این نرم افزار در یادگیری ماشین و داده کاوی و متن کاوی و آنالیز پیش بینی و آنالیز تجارت استفاده می شود. این نرم افزار هم در محیط های صنعتی و هم دانشسگاهی مورد استفاده قرار می گیرد. گفته می شود این نرم افزار ۹۹ درصد راه حل های آنالیزی پیشرفته را بدون هیچ زحمت کدنویسی برای محقق داده کاوی فراهم می کند. این نرم افزار با زبان برنامه نویسی جاوا نوشته شده است. تمامی عملیات در نرم افزار ریپیدماینر توسط پنجره گرافیکی انجام می شود. عملیات workflow در ریپیدماینر به نام فرایند process شناخته می شوند که شامل چندین عملگر operator تشکیل می شوند. هر عملگر یک وظیفه را در فرایند انجام می دهد. این نرم افزار قابلیت ارتباط با نرم افزارهای داده کاوی دیگر مانند R و weka را دارد. RapidMiner. قوی ترین و آسان ترین نرم افزار برای آنالیز داده ها به صورت عددی و نمایش گرافیکی نتایج حاصل از تحلیل داده ها است. هم چنین این نرم افزار قابلیت ورود کدهای داده کاوی از سایر برنامه ها مانند R و Python را دارد و می تواند از داده های سایر نرم افزار های داده کاوی مانند Microsoft Excel, Oracle, Microsoft Access, IBM DB, Microsoft SQL Server, Teradata نیز استفاده کند. در این نرم افزار سعی گروه توسعه دهنده، بر آن بوده است که تا حد امکان تمامی الگوریتم های رایج داده کاوی و هم چنین یادگیری ماشین پوشش داده شوند. به گونه ای که حتی این امکان برای نرم افزار فراهم شده است تا بتوان سایر ابزارهای متن باز داده کاوی را نیز به آن الحاق نمود. از این رو این نرم افزار گاهی در کنار پایتون و R از پرکاربردترین نرم افزار های مورد استفاده تحلیلگران داده میباشد.

## ۳ فصل سوم

### ۱-۳ حل مسئله پیش بینی انرژی با رگرسیون از طریق کتابخانه scikit-learn :

ابتدا موارد مورد نیاز از کتابخانه ها را فراخوانی میکنیم سپس فایل CSV شامل داده های مسئله را به کمک کتابخانه پانداس فراخوانی کرده و در energy ذخیره میکنیم. اکنون با دستور shape میتوانیم تعداد سطرها و ستون ها را مشاهده کنیم.

```
In [1]: import numpy as np
import pandas as pd

from sklearn import datasets
from sklearn import linear_model

from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

```
In [2]: energy = pd.read_csv("Desktop/energydatacomplete.csv")
energy.shape
```

```
Out[2]: (19735, 29)
```

در اینجا با دستور `head()` میتوانیم با دادن مقدار 5 (تعداد سطرهای دلخواه) شمایه کلی جدول داده را که متشکل از 29 ستون می باشد مشاهده نماییم.

ستون اول یا `appliances` داده ای است که در مسئله به کمک رگرسیون باید پیشبینی شود. در ستون های بعدی اطلاعات دما و رطوبت را در نقاط مختلف خانه داریم و نهایتا به دو متغیر تصادفی `rv1,rv2` میرسیم که به دلیل تصادفی بودن تاثیر مثبتی روی پیشبینی هدف یعنی `appliances` ندارند لذا در ادامه اقدام به حذف آنها میکنیم.

```
In [3]: energy.head(5)
```

```
Out[3]:
```

	date	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	...	T9	RH_9	T_out	Press_mm_hg	RH_out	Windspe
0	2016-01-11 17:00:00	60	30	19.89	47.598887	19.2	44.790000	19.79	44.730000	19.000000	...	17.033333	45.53	6.600000	733.5	92.0	7.0000
1	2016-01-11 17:10:00	60	30	19.89	46.693333	19.2	44.722500	19.79	44.790000	19.000000	...	17.066667	45.56	6.483333	733.6	92.0	6.6667
2	2016-01-11 17:20:00	50	30	19.89	46.300000	19.2	44.626667	19.79	44.933333	18.926667	...	17.000000	45.50	6.366667	733.7	92.0	6.3333
3	2016-01-11 17:30:00	50	40	19.89	46.066667	19.2	44.590000	19.79	45.000000	18.890000	...	17.000000	45.40	6.250000	733.8	92.0	6.0000
4	2016-01-11 17:40:00	60	40	19.89	46.333333	19.2	44.530000	19.79	45.000000	18.890000	...	17.000000	45.40	6.133333	733.9	92.0	5.6667

5 rows x 29 columns

با دستور `loc`، غیر از دو ستون آخر بقیه ستون هارا در جای خود قرار داده و همه ستون هارا به `X` نسبت میدهیم تا با استفاده از این دیتا ها ستون `Y` را پیش بینی کنیم:

```
In [4]: X = energy.loc[:, 'lights':'Tdewpoint']
        Y = energy.Appliances
        X
```

```
Out[4]:
```

	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	RH_4	T5	...	T8	RH_8	T9	RH_9
0	30	19.890000	47.598887	19.200000	44.790000	19.790000	44.730000	19.000000	45.566667	17.166667	...	18.2000	48.900000	17.033333	45.5300
1	30	19.890000	46.693333	19.200000	44.722500	19.790000	44.790000	19.000000	45.992500	17.166667	...	18.2000	48.863333	17.066667	45.5800
2	30	19.890000	46.300000	19.200000	44.626667	19.790000	44.933333	18.926667	45.890000	17.166667	...	18.2000	48.730000	17.000000	45.5000
3	40	19.890000	46.066667	19.200000	44.590000	19.790000	45.000000	18.890000	45.723333	17.166667	...	18.1000	48.590000	17.000000	45.4000
4	40	19.890000	46.333333	19.200000	44.530000	19.790000	45.000000	18.890000	45.530000	17.200000	...	18.1000	48.590000	17.000000	45.4000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19730	0	25.566667	46.560000	25.890000	42.025714	27.200000	41.163333	24.700000	45.590000	23.200000	...	24.7000	50.074000	23.200000	46.7900 2
19731	0	25.500000	46.500000	25.754000	42.080000	27.133333	41.223333	24.700000	45.590000	23.230000	...	24.7000	49.790000	23.200000	46.7900 2
19732	10	25.500000	46.596667	25.628571	42.768571	27.050000	41.690000	24.700000	45.730000	23.230000	...	24.7000	49.680000	23.200000	46.7900 2
19733	10	25.500000	46.990000	25.414000	43.036000	26.890000	41.290000	24.700000	45.790000	23.200000	...	24.6625	49.518750	23.200000	46.8175 2
19734	10	25.500000	46.600000	25.264286	42.971429	26.823333	41.156667	24.700000	45.963333	23.200000	...	24.7360	49.736000	23.200000	46.8450 2

19735 rows x 25 columns

به منظور آموزش و الگو برداری به کمک تابع `train_test_split` داده ها را به صورت تصادفی و با میزان 70 به 30 به 4 دسته برای تعلیم دادن و تست کردن جدا میکنیم.

همانطور که مشاهده میشود تعداد 13814 سطر جهت تعلیم و 5921 داده جهت تست به صورت تصادفی جدا شده است. (تست سایز ثابت به منظور عوض نشدن مجموعه تعلیم و تست میباشد)

پس از آن مدل خطی را رگرسیون قرار داده و آن را بر روی داده های تعلیمی سازگار میکنیم. سپس متغیر جدیدی به نام `Y_pred` به معنای پیش بینی مقادیر تست ایجاد میکنیم و مقادیر پیش بینی مدل رگرسیون را در آن میریزیم.

```
In [10]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.30)
print(X_train.shape , Y_train.shape)
print(X_test.shape , Y_test.shape)

(13814, 25) (13814,)
(5921, 25) (5921,)
```

```
In [11]: model = linear_model.LinearRegression()
model.fit(X_train, Y_train)
```

```
Out[11]: LinearRegression()
```

```
In [14]: Y_pred = model.predict(X_test)
```

در آخرین مرحله جهت امتحان کردن میزان درستی پیش بینی مدل، میزان کمترین خطای مطلق (MAE) و میزان خطای کمترین مربعات (MSE) را محاسبه میکنیم.

طبق اعداد به دست آمده انتخاب مدل رگرسیون پیشبینی خوبی از داده های هدف به ما داده است اما این پیشبینی لزوماً بهترین پیشبینی نیست و میزان خطای نسبتاً زیادی دارد.

```
In [19]: mean_absolute_error(Y_test, Y_pred)
```

```
Out[19]: 52.114307226525746
```

```
In [20]: from math import sqrt
sqrt(mean_squared_error(Y_test, Y_pred))
```

```
Out[20]: 92.01511766173867
```

## جمع بندی

پایتون ابزاری قدرتمند برای تحلیلگران داده است. از این رو تمامی ویژگی های آن را نام برده و دلایل برتری و محبوبیت آن نسبت به سایر نرم افزار ها را به وضوح مشخص کردیم. بدیهی است که حل تمامی مسائل تحلیلی، تشخیص الگوها و بصری سازی نتایج تنها به کمک پایتون میسر نیست و تحلیلگران باید به شناخت عمیق و استفاده از سایر ابزار های قدرتمند تحلیل داده نظیر R، Rapidminer و... جهت استفاده بهینه از منابع خود بپردازند.



## منابع و مراجع:

- سایت [Problem Solving with Python](#)
- سایت [Linear Regression and Regularisation | Kaggle](#)
- دوره یادگیری ماشین و پایتون استاد شیرافکن

پایان